Using Early View Patterns to Predict the Popularity of YouTube Videos

Henrique Pinto, Jussara M. Almeida, Marcos A. Gonçalves

Computer Science Department
Universidade Federal de Minas Gerais, Brazil
{hpinto, jussara, mgoncalv}@dcc.ufmg.br

ABSTRACT

Predicting Web content popularity is an important task for supporting the design and evaluation of a wide range of systems, from targeted advertising to effective search and recommendation services. We here present two simple models for predicting the future popularity of Web content based on historical information given by early popularity measures. Our approach is validated on datasets consisting of videos from the widely used YouTube video-sharing portal. Our experimental results show that, compared to a state-of-the-art baseline model, our proposed models lead to significant decreases in relative squared errors, reaching up to 20% reduction on average, and larger reductions (of up to 71%) for videos that experience a high peak in popularity in their early days followed by a sharp decrease in popularity.

Categories and Subject Descriptors

C.4 [**Computer Systems Organization**]: Performance of Systems - *Measurement techniques*; H.3.5 [**Information Storage and Retrieval**]: Online Information Services - *Web-based services*

General Terms

Experimentation, Measurement

Keywords

popularity prediction; YouTube; regression models

1. INTRODUCTION

The increasing popularity of Web 2.0 applications brings along an enormous and ever growing amount of user generated content. Take for instance the YouTube video sharing system, which often figures among the top 3 most popular applications on the Web [1]. It has been reported that YouTube users upload 72 hours of video per *minute*[1], and that the total amount of content uploaded to YouTube in 60 days is equivalent to all content that would have been broadcasted for 60 years, without interruption, by NBC, CBS and ABC altogether [9]. Given such staggering content upload rate, it is unsurprising that the distribution of popularity across different contents on YouTube, as well as on other Web applications, is very uneven: most content garners very little attention whereas a small amount of it attracts millions of views [4, 16].

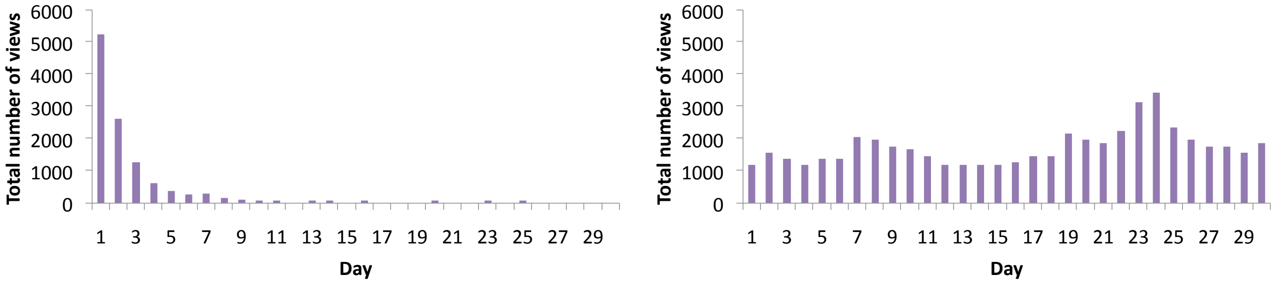
In this context, predicting content popularity is of great importance to support and drive the design and management of various services. For example, in online marketing, the information about the expected future popularity of a certain type of content can be useful for planning advertising campaigns and estimating costs. Accurately predicting content popularity is also key to support effective information services (e.g., recommendation and searching services) [3]. Popularity prediction can help identifying possible bottlenecks due to poor recommendation and search engines and improving the quality of such services by extending current result ranking strategies to take the estimated future popularity into account [8].

There have been recent efforts towards building models to predict the popularity of online content using various techniques such as reservoir computing [17], stochastic models of user behavior [12] and biology-inspired survival analysis techniques [11]. Towards that goal, Szabo and Huberman first observed that the log-transformed long-term popularity of a given content is strongly correlated with its early popularity [15]. Based on this observation, they proposed a simple model that predicts the total number of views (i.e., the popularity) of a piece of content at a target date t_t, based on a linear function of its *total number of views* at an earlier *reference date* t_r ($t_r < t_t$). That is, their model states that the future popularity is related to the early one by a constant factor α, which depends only on the target and reference dates and not on any specific information of the content itself, and can be fitted by linear regression.

While the reported results produced by the Szabo and Huberman (S-H) model are reasonably accurate, especially given its simplicity, the model does have shortcomings. In particular, two different pieces of content may have very similar popularity at the same reference date and yet exhibit distinct popularity behaviors thereafter. Take, for instance, the two real YouTube videos whose popularity curves, in terms of the daily number of views for the first 30 days, are shown in Figure 1. After 7 days since upload, both videos have very similar total number of views: $10,665$ for video A,

[1] `http://www.youtube.com/t/press_statistics`

WSDM'13, February 4–8, 2013, Rome, Italy.



(a) Video A: 10,665 at t_r=7; 12,060 views at t_t=30

(b) Video B: 10,070 views at t_r=7; 51,851 views at t_t=30

Figure 1: Number of views per day during the first 30 days of life of two YouTube videos

and 10,070 for video B. Thus, the S-H model would predict, using data from these days (t_r=7), that both videos would have similar total number of views on the 30th day since upload (t_t=30). Yet, they end up with very different total popularities: whereas Figure 1-a) shows a final aggregate popularity that barely exceeds 12,000 views, the video in Figure 1-b) ends up attracting more than 50,000 views during the same period. This example illustrates that different YouTube videos (and online content in general) may experience very different *popularity evolution patterns*. Indeed, Crane and Sornette have distinguished four different popularity evolution patterns among YouTube videos [5], which are explained in terms of a combination of endogenous (i.e., user interactions within the system) and exogenous factors (i.e., external events), as further discussed in Section 2.

Thus, we here investigate whether the *view patterns* of a video during its early days in the system up to a reference date t_r lead to more accurate predictions of its total number of views at a target date t_t, in comparison to simply using the total view count up to t_r (as done by the S-H model).

We propose two new popularity prediction models. The first is a multivariate linear regression model, the ML model, that, building on the S-H model, incorporates information about historical patterns. It is motivated by the observation that if one is monitoring the number of views of a video for t_r days, these days are not all equally important to predict its future popularity. Thus, like the S-H model, our model relies only on the number of views up to a reference date t_r, but it assigns different weights to each monitored day, thus allowing us to distinguish between videos with roughly the same number of views at t_r but very different popularity curves (as in Figure 1).

Our second prediction model, referred to as MRBF, is an extension of the ML model that aims to exploit the different popularity patterns a video can follow in a more explicit way. By measuring the similarity of a video and known examples from the training set, and changing the prediction based on this information, it is able to adapt to the different patterns in a more direct way, leading to further improvements in prediction accuracy.

We evaluate our models, comparing them against the S-H model, on two datasets of YouTube videos. One consists of popular videos that appear on the world-wide top lists kept by YouTube, and the other contains videos sampled according to a random procedure [6]. For each such dataset, we also measure how well each model performs, in terms of prediction error, for various values of t_r and t_t as well as for videos that exhibit different popularity evolution patterns (as defined in [5]). We find that our models significantly improve prediction accuracy over the S-H model. For instance, when predicting the popularity of a video at the end of a month (t_t=30) using only data from the first week (t_r=7), the ML model leads to reductions in prediction error of 13% and 15%, on average, for videos on the top and random datasets, respectively. The MRBF model further reduces the prediction errors by 6% on each dataset, leading to total reductions in error over the S-H baseline of 19% and 21% for the top and random datasets, respectively.

Our models exploit the same data used by the S-H model, with little extra cost in practice (see discussion in Section 3.3). We find that such accuracy improvements can provide real benefits to services that exploit future popularity estimates. Moreover, we find that for specific patterns of views, our models perform especially well. For example, for videos in the random dataset that exhibit very high peaks followed by sharp decreases in popularity (as in Figure 1-a), the MRBF model leads to an accuracy gain of 71% over the S-H model.

Motivated by a question left open by Szabo and Huberman [15], we also evaluated whether building specialized models for each YouTube video category leads to more accurate popularity predictions. However, we found that this type of model specialization has little impact on prediction error (except in a single case), because the videos in most categories follow the same general popularity evolution patterns,. Thus a global model, trained on videos of all categories, can adequately predict the popularity of videos in any category.

The rest of this paper is organized as follows. Section 2 discusses related work, whereas Section 3 formally presents both the S-H and our new popularity prediction model. Our evaluation methodology and main results are discussed in Sections 4 and 5, respectively. Section 6 concludes the paper.

2. RELATED WORK

There have been several efforts towards analyzing the popularity of online content. For example, Cha *et al.* studied the popularity cycle of YouTube videos [4], finding that they have, on average, long life times, and that around 80% of the videos watched on a day are older than one month, although the most popular video tends to be one that has been recently uploaded. Figueiredo *et al.* characterized the popularity evolution patterns of YouTube videos and studied the impact of different types of referrers on such patterns [6], whereas Rodrigues *et al.* analyzed the characteristics of groups of duplicates of the same YouTube video, finding that they often have different popularities [13].

Crane and Sornette analyzed the popularity evolution patterns of YouTube videos, identifying four main classes, which they explained in terms of endogenous and exogenous effects [5]. According to them, the majority of the videos experience no marked peak in popularity: they either attract little attention or experience some popularity fluctuation that can be explained through a simple stochastic process. They referred to these videos as *Memoryless.* In contrast, the other videos experience bursts in popularity, being further categorized into *Viral*, *Quality* or *Junk* videos. *Viral* videos experience a popularity peak that emerges through a word-of-mouth epidemic-like internal propagation process. Their popularity increases slowly, up to a peak, decreasing also slowly afterwards. *Quality* videos experience a very sudden peak in popularity, possibly due to some external event (such as being featured on the first page of YouTube), and a slow decay afterwards as users propagate the videos among themselves. *Junk* videos also experience a burst of popularity, but, in contrast, they do not spread through the social network and thus their popularity drops quickly afterwards. The videos in Figures 1-(a) and 1-(b) are, respectively, in the *Junk* and *Memoryless* classes. The authors also offered an alternative classification procedure of videos into *Viral*, *Quality* and *Junk* based on the fraction of views on the peak day. In Section 5.1, we analyze the accuracy of the popularity prediction models for videos in each of these classes.

Popularity prediction has also been the goal of a few recent studies. Lerman and Hogg predicted the popularity of Digg[2] stories by modeling user behavior explicitly as a stochastic process [12]. They considered aspects of the user interface of the system, as well as the social dynamics of voting due to the *friends* feature. They model the user interaction with the system in terms of actions, such as visiting the home page, clicking on the "next" link to go to the second page, visiting the friends page, voting on a news item and others, and considered how these actions influence popularity. They found that there are two main components for explaining popularity evolution: a story's inherent *quality* and *social influence.* While their model produced good results for Digg content, its effectiveness on a system like YouTube is questionable. This is because YouTube has many more internal features and mechanisms, allowing a richer variety of user interactions, which can ultimately impact popularity. Thus, modeling YouTube user behavior and its impact on content popularity becomes much more complex. In particular, in order for a user behavior model such as the one proposed on [12] to perform well on a system like YouTube, external factors (e.g., search and links in other web sites), which significantly influence popularity [6], should also be taken into account and explicitly modeled.

Wu *et al.* [17] proposed a model based on *reservoir computing*, a special type of neural network, to predict the near future popularity of videos based on popularity data from the previous days. However, their strategy can be affected by randomization effects, as the hidden layer weights are set to randomly chosen *fixed* values, and it takes a number of iterations until the outputs reflect the data instead of the random weights. Moreover, the proposed model does not outperform simpler strategies, such as the Szabo-Huberman model, in large datasets, as reported by the authors.

Lee *et al.* [11] proposed a model to predict the likelihood that a thread in discussion forums will still be popular in the future. They used biology-inspired *survival analysis* techniques that take certain "risk factors" as input. Examples of such factors are: the total number of comments and the time between the thread's creation and the first comment. Their model is built on the assumption that measures of the actual popularity are not available, which may be too restrictive for popularity prediction on such systems. On YouTube, in particular, this is unrealistic, as the system makes such data publicly available. Moreover, we found that the risk factors used by their model are highly correlated with the number of views of the videos in our datasets, bringing no improvements to our models.

Yin *et al.* [18] proposed a model that predicts the expected popularity ranking of items. In many systems, users can express positive or negative opinions about items (such as "like/dislike" on YouTube). Through an empirical study, they noticed that users display two personalities: they are either "Conformers", voting according to the opinion of the majority, or "Mavericks", who vote in disagreement with the majority. They argued that an individual user exhibits both of these personalities to different degrees, and that when a user votes on a specific item, one of these personalities prevails. Based on this, they created a model for user behavior and used the early votes to develop a function to rank items based on their potential popularity. They evaluated their proposed ranking function on a dataset collected from JokeBox[3], a popular iPhone application that allows users to post and vote on jokes. The proposed method outperformed many other strategies when tasked with predicting the top-k most popular jokes. One disadvantage of their method, though, is that it requires full information about users' votes in order to build user profiles. This information is not publicly available in many systems, including YouTube, either due to scale issues, or privacy issues, or both.

In contrast to these efforts, Szabo and Huberman analyzed the popularity of YouTube videos and Digg stories, noting that the long term popularity of online content is strongly correlated with its early popularity in a logarithmic scale [15]. Based on that observation, they proposed a very simple linear popularity prediction model. Given its simplicity and promising results for different applications, we here choose the Szabo and Huberman model, which is further described in Section 3.2, as baseline for evaluating our new strategies.

3. POPULARITY PREDICTION MODELS

We start our discussion of the popularity prediction models by presenting the criterion adopted for evaluating their performance (Section 3.1). We then present the baseline S-H model (Section 3.2) as well as our ML (Section 3.3) and MRBF (Section 3.4) models, discussing how each model tries to optimize the adopted performance criterion. The goal of all the prediction models is to predict the future popularity,

[2] Digg (`http://digg.com`) is a news aggregator website where users can post links to stories, votes and comments.

[3] `http://itunes.apple.com/us/app/all-in-1-joke-box-no-adsunlimited/id363494433?mt=8`

in terms of *total number of views*, of a video at t_t, given data from the first t_r days ($t_r < t_t$).

3.1 Performance Criterion

As in [15], we use the Relative Squared Error (RSE) to evaluate the performance of the prediction models. Let $N(v,t)$ be the total number of views video v receives up to day t ($N(v,0)=0$), and $\hat{N}(v,t_r,t_t)$ be the total number of views *predicted* for v at target date t_t based on data from the first t_r days. The RSE for this prediction is given by:

$$RSE = \left(\frac{\hat{N}(v,t_r,t_t)}{N(v,t_t)} - 1\right)^2 \tag{1}$$

For a collection C of videos, the mean Relative Squared Error (mRSE) is defined as the arithmetic mean of the RSE values for all videos in C[4], that is:

$$mRSE = \frac{1}{|C|} \cdot \sum_{v \in C} \left(\frac{\hat{N}(v,t_r,t_t)}{N(v,t_t)} - 1\right)^2 \tag{2}$$

We here adopt the RSE, instead of the absolute quadratic error because, for various services for which popularity prediction is useful (e.g., search engine and content recommendation), relative errors tend to be more relevant and meaningful than absolute ones, particularly given the great variability in popularity across different contents [4].

3.2 Szabo-Huberman (S-H) Model

Szabo and Huberman [15] observed a high linear correlation between the log-transformed early and future popularities of online content up to a normally distributed noise. Based on this observation, they expressed the future popularity of a piece of content v as:

$$\hat{N}(v,t_r,t_t) = \alpha_{t_r,t_t} \cdot N(v,t_r) \tag{3}$$

where parameter α_{t_r,t_t} is independent of the video v. This means that, for fixed t_r and t_t, the future popularity of v is related to its early one by a constant factor.

For any given pair (t_r,t_t), we can compute the optimal value for α_{t_r,t_t} in a given training set C of videos, by plugging the expression for $\hat{N}(v,t_r,t_t)$ in Equation 2, taking its derivative and equating it to zero. This procedure leads to:

$$\alpha_{t_r,t_t} = \frac{\sum_{v \in C} \frac{N(v,t_r)}{N(v,t_t)}}{\sum_{v \in C} \left(\frac{N(v,t_r)}{N(v,t_t)}\right)^2} \tag{4}$$

Computing this optimal value for the model parameter can be done in $O(n)$ on a training set with n examples.

3.3 Multivariate Linear (ML) Model

The S-H model uses as input the total number of views of a video up to day t_r. We here consider that, instead of having only one number, we sample the total number of views at regular intervals up to the same t_r date, thus computing the number of views received *per sampling interval*, which can be seen as popularity *deltas* up to t_r. Unless otherwise stated, we here assume daily samplings for simplicity, although the model itself makes no assumption on the sampling rate.

Our proposed multivariate linear (ML) model predicts the popularity of a video at t_t as a linear function of these popularity deltas. The linear assumption is reasonable given the strong linear correlation between early and future popularities observed in [15]. Yet, the model is more powerful than the S-H model as it allows the assignment of different "weights" to each sampling interval.

More formally, let $x_i(v)$ be the number of views received by video v on the i-th day since its upload ($x_i(v) = N(v,i) - N(v,i-1)$). The *feature vector* $X_{t_r}(v)$ is defined as:

$$X_{t_r}(v) = (x_1(v),\, x_2(v),\, \ldots, x_{t_r}(v))^T$$

and we estimate the popularity of the video v at t_t as:

$$\hat{N}(v,t_r,t_t) = \Theta_{(t_r,t_t)} \cdot X_{t_r}(v) \tag{5}$$

where $\Theta_{(t_r,t_t)} = (\theta_1,\, \theta_2,\, \ldots,\, \theta_{t_r})$ is the vector of *model parameters* and depends only on t_r and t_t.

Given a training set C, t_r and t_t, we can compute the optimal values for the elements of $\Theta_{(t_r,t_t)}$ as the ones that minimize the mRSE on C, i.e.:

$$\underset{\Theta_{(t_r,t_t)}}{\arg\min} \frac{1}{|C|} \sum_{v \in C} \left(\frac{\Theta_{(t_r,t_t)} \cdot X_{t_r}(v)}{N(v,t_t)} - 1\right)^2 \tag{6}$$

The hypothesis is that $\hat{N}(v,t_r,t_t)$ is a linear function, and $N(v,t_t)$ a scalar. Thus, it follows that:

$$\frac{\Theta_{(t_r,t_t)} \cdot X_{t_r}(v)}{N(v,t_t)} = \Theta_{(t_r,t_t)} \cdot \left(\frac{X_{t_r}(v)}{N(v,t_t)}\right)$$

Let $X_v^* = \frac{X_{t_r}(v)}{N(v,t_t)}$. We express the optimization problem as:

$$\underset{\Theta_{(t_r,t_t)}}{\arg\min} \frac{1}{|C|} \sum_{v \in C} \left(\Theta_{(t_r,t_t)} \cdot X_v^* - 1\right)^2, \tag{7}$$

which is an ordinary least squares (OLS) problem that can be solved via a singular value decomposition [2], with complexity $O(np^2)$, where n is the number of training examples and p the number of model parameters. We assume that $n \geq p$.

The S-H model is a special case of this multivariate linear model, with the added restriction that $\theta_1 = \theta_2 = \cdots = \theta_{t_r}$. As such, the aggregate results, in terms of mRSE, provided by our model should be always either equal or better than those generated by the S-H model, although this is not guaranteed for individual videos. One possible drawback of our ML model, though, is that the number of parameters is not fixed, but increases linearly with t_r. We believe, however, that this would not be an issue in practice because: (1) one would usually not observe the popularity for very long before the prediction; and (2) if the total number of samples, and thus the number of parameters, is too large, we can reduce it by using a larger sampling interval (for example, weekly instead of daily).

3.4 MRBF Model

By assigning different weights to different days in the observed history of the video, the ML model is able to capture some information about the popularity evolution patterns of videos. However, it is still limited by using a single set of parameters for all videos. The different popularity patterns display very different behaviors, and thus it is likely

[4] We note that the RSE is not defined for videos that have zero views at the target date. However, we find that only a very small fraction (below 1.5%) of the videos in our datasets have zero views at all target dates we considered. Thus, we disregard those videos when computing mRSE values.

that exploring particular aspects of each pattern can lead to improved prediction accuracy.

One possible approach for solving that problem is to create different, specialized models for each pattern. Applying those models, though, is problematic, because it requires choosing the most appropriate model for the video for which we want to predict the future popularity, which ultimately implies that we have to be able to predict, a priori, based solely on the early view measures, the overall popularity pattern the video will follow. This is a hard, still open problem.

We opt here for a different approach. Instead of building a classifier for predicting the popularity pattern of videos, we build a new model that takes into account the similarity (in terms of early views, up to t_r) between the video and known examples from the training set, and uses this similarity to adapt the popularity prediction. This new model extends the ML model using additional features that measure the similarity between the video for which one wishes to predict the future popularity and some specific examples from the training set. Using these extra features, we can tweak the popularity prediction to better capture some particular aspect of certain types of videos. In essence, we still keep a single-parameter-set ML model, but alter that prediction based on the similarity to certain videos. Assuming that the videos from the training set used for this computation encompass the various popularity patterns, we are (indirectly) incorporating information about the pattern of the video for which we are predicting the future popularity, and adapting the prediction model to it.

For measuring the similarity between videos, we use Radial Basis Functions (RBFs). A radial basis function is a real-valued function whose value depends only on the distance between its inputs and a given point, the *center* [7]. Given a training set video v_c, we create a Gaussian RBF with it as the center to capture its similarity with a target video v as follows:

$$RBF_{v_c}(v) = \mathrm{e}^{\left(-\frac{||X(v)-X(v_c)||^2}{2\cdot\sigma^2}\right)}, \qquad (8)$$

where σ is a parameter and $X(v)$ is the ML model feature vector for the video v.

We select an uniform random sample of examples from the training set and use these as centers for RBF features. For each video v, we then compute $RBF_{v_c}(v)$ and use it as one of the features in the prediction model. We call this model the MRBF model, and formally define it as:

$$\hat{N}(v, t_r, t_t) = \underbrace{\Theta_{(t_r,t_t)} \cdot X(v)}_{\text{ML Model}} + \underbrace{\sum_{v_c \in C} \omega_{v_c} \cdot RBF_{v_c}(v)}_{\text{RBF Features}} \qquad (9)$$

where C is the set of training set examples chosen as centers and ω_{v_c} is the model weight associated with the RBF feature for v_c used for prediction purposes.

In order to use the MRBF model, we must first set appropriate values for α, σ and the number of training set examples chosen as centers. There is no a priori best value for any of these parameters. We experimented with many values for each of them and chose to use the values that provided the lowest prediction error in a cross-validation set.

Once these parameters are set, training the MRBF model consists of finding optimal values for the Θ and ω_{v_c} model parameters. In order to do so, we could perform regular linear regression, as we did for the ML model. Notice that the MRBF model as defined in Equation 9 is mathematically equivalent to:

$$\hat{N}(v, t_r, t_t) = \Theta^*_{(t_r,t_t)} \cdot X^*_{t_r}(v) \qquad (10)$$

where Θ^* is the Θ vector with the w_{v_c} parameters appended to it and $X^*(v)$ is the $X(v)$ vector with the values of the corresponding RBF functions appended to it – i.e., the RBF features can be simply treated as additional features in the original feature and parameter vectors. Equation 10 is in exactly the same format as Equation 5 that describes the ML model. Thus, it can be solved using the same OLS technique. However, due to the extra features, we are at increased risk of overfitting our training set. To reduce this problem, we opted to use Ridge regression [7] instead.

Ridge regression is a regression method that works in a very similar manner to the ordinary least squares (OLS) method, but additionally imposes a penalty on the sizes of the coefficients. In our ML model, which is based on OLS regression, we expressed the optimization problem as defined in Equation 7. This equation implies that we want to find the value for the parameter vector $\Theta^*_{(t_r,t_t)}$ that minimizes the prediction error. For Ridge regression, we add an extra term to the optimization problem that penalizes solutions where the norm of $\Theta^*_{(t_r,t_t)}$ is large:

$$\underset{\Theta^*_{(t_r,t_t)}}{\arg\min} \frac{1}{|C|}\left(\sum_{v\in C}\left(\Theta^*_{(t_r,t_t)} \cdot X^*_v - 1\right)^2\right) + \alpha \cdot ||\Theta^*_{(t_r,t_t)}||^2$$

where α is a parameter that controls how large the penalty for larger coefficients can be. The effect of penalizing solutions with large norms is that we avoid overfit, as usually solutions with smaller coefficients tend to have higher generalization power [7]. We tested this by comparing the errors of the MRBF model in training and test sets. If overfit occurred, we would expect the error in the training set to be much lower than the error in the test set. We found, however, that for reasonable values of α, the error in the test set is very close to the error on the training set.

4. DATASETS

We evaluate our ML and MRBF models, comparing them against the S-H model, on two datasets of YouTube videos. For both datasets, we choose to focus on videos with at least 30 days in the system, varying t_t from 30 to 100. Our datasets are:

- *Top*: YouTube maintains various per-country and world-wide top lists (e.g., most viewed, most commented, most responded videos), allowing users to browse them in different time scales (top of the day, week, month and top of all time). Each such top list contains one hundred videos. This dataset contains videos from all world-wide top lists provided by YouTube. 27,212 videos were collected through this procedure.

- *Random*: we also would like to evaluate the prediction models on a random sample of YouTube videos. As the system does not provide a means to collect a truly random sample of all its videos, alternative sampling procedures must be adopted. We here use a dataset consisting of videos collected in a procedure that is based on random topics. First, 30,000 entities (i.e., words and proper names) were randomly selected from the Yago lexical ontology [14]. Each such entity was then fed as query to the YouTube search engine, and the first result was collected. Overall, this resulted in 24, 484 unique videos.

For both datasets, the view count data for each video was extracted from the video statistics frame provided by YouTube, where it is shown in graph form, plotted using the Google Charts API[5]. The API call to plot the graph was intercepted and the plotted points were collected. Each graph contains at most 100 points. This means that, for videos with at most 100 days of life, the system maintains complete information about the daily view counts, whereas for older videos some daily view counts are missing. For example, if a video is in the system for 300 days, then the graph will be plotted with 100 points, each point corresponding to a 3-day interval. For such videos, we used linear interpolation to infer the missing points. We note that the interpolation may impact the prediction results. However, we repeated our experiments considering only videos with complete information, finding that the qualitative results are not affected, and that the mRSE values for models trained on the interpolated data are actually slightly higher than those obtained with models trained only on videos that do not require interpolation, for the S-H, the ML and the MRBF models. Nevertheless, we choose to present, in the next section, results that include videos for which interpolation was used, due to the small number of videos with complete information in our datasets.

We noticed that many videos in YouTube do not have statistics available. Since these statistics represent key input data for our prediction experiments, we removed from the datasets these videos with missing information. We also removed videos that were in the system for less than 30 days. After applying these filters, our Top and Random datasets were left with 5, 834 and 16, 123 videos, respectively.

5. EXPERIMENTAL RESULTS

In this section, we discuss the main experimental results of the S-H model and our new ML and MRBF model. These results were produced using 10-fold cross validation. That is, each dataset was randomly divided into 10 equal-sized folds. 9 folds were used as training set from which model parameters are learned, and the other fold was used as test set. We repeated the same process 10 times, using a different fold as test set each time. We report average results across all 10 test sets along with corresponding 95% confidence intervals (except in the charts, where the intervals are omitted to improve readability).

We divide our experimental analysis in three parts. First, in Section 5.1, we analyze in details the performance of the S-H model and of our ML model, explain how and why the latter outperforms the baseline. Then, in Section 5.2, we compare the ML model to the MRBF model, discussing what is the impact in popularity prediction of adding the RBF features to the ML model. Finally, in Section 5.3 we discuss our experiment in creating specialized models for each YouTube category.

5.1 ML and S-H Model Results

Figure 2-a) shows mRSE values for both S-H and our new ML models when predicting the popularity for a target date t_t equal to 30 days and various values of reference date t_r, for videos in the random dataset. Note that if t_r is too small, there is very little extra information that our ML model can benefit from, and thus both models perform very similarly. On the other hand, if almost the total popularity history is used (i.e., t_r is close to t_t), the errors are so small that there is little space for improvements. In any case, predicting popularity in this latter scenario is of little practical use. More importantly, Figure 2-a) shows that, outside these two extreme scenarios, the ML model leads to significant mRSE reductions over the S-H model, with gains in precision of over 20%. For example, if we use $t_r = 12$ days, the mRSE produced by the ML and by the S-H models are 0.081 ± 0.009 and 0.104 ± 0.008, respectively, leading to an error reduction of 22%, on average. For smaller values of t_r, the gains are reduced but still significant (e.g., 15% for t_r equal to 7 days).

Figure 2-b) and 2-c) show similar patterns for results produced for the top dataset and $t_t = 30$, as well as for the random dataset and $t_t = 100$. In the latter case, the reductions in mRSE values are even larger. For instance, if we use $t_r = 40$ to predict for $t_t = 100$, the mRSE produced by the ML and S-H models are 0.1112 ± 0.0024 and 0.1679 ± 0.0025, respectively, yielding a 33% error reduction on average.

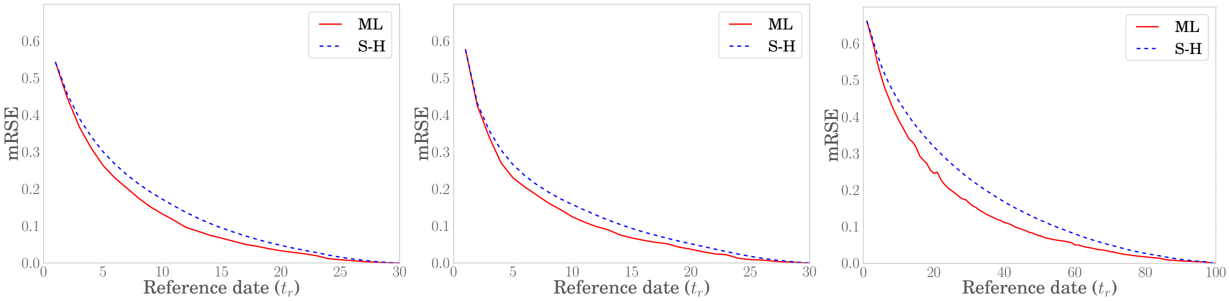
To further understand the performance of the ML model, we separately analyze the mRSE values produced by both models for videos with different popularity evolution patterns. Towards that goal, we group the videos in our datasets into the four classes identified by Crane and Sornette [5], namely *Memoryless*, *Viral*, *Junk* and *Quality*, following the approach adopted in [5, 6]. To identify Memoryless videos, a Chi-Square test was applied to determine whether the time series describing the popularity evolution of each video follows a Poisson process. Videos for which the test failed were further classified into Viral, Junk and Quality based on the fraction f of views on the most popular day: videos with $f \leq 35\%$ were classified as Viral, videos with $f > 65\%$ as Junk, and the others were classified as Quality.

Table 1 shows the mRSE results produced by all three models, for both random and top dataset, considering t_r=7 and t_t=30. Results for other values of t_r and t_t are qualitatively similar, being thus omitted. Aggregate as well as per-class results are shown[6]. In each case, results for the most accurate model are shown in bold (including statistical ties at 95% confidence level). The table also shows the model equations for the S-H and ML models, indicating the optimal parameters found during training (parameters for the MRBF models were omitted for space). We will first analyze the results for the ML and S-H models, and defer the analysis of the MRBF results until Section 5.2.

The overall mRSE reductions achieved with our ML model over the baseline, across all classes, are 15% and 13% for the random and top datasets, respectively. It produces significant error reductions for videos in most classes while reaching statistical ties in a few others. The gains are especially large for *Junk* videos in the random dataset (69% on average) (see discussion below). Such a large reduction in pre-

[5] http://code.google.com/apis/chart/

[6] These results are produced by one global S-H model, one global ML model, and one global MRBF model, built from the complete training set.



(a) t_t = 30 days, random dataset

(b) t_t = 30 days, top dataset

(c) t_t = 100 days, random dataset

Figure 2: Model Prediction Errors (mRSE) as Function of Reference Date t_r for Various Target Dates t_t

Table 1: Prediction Errors for S-M, ML, and MRBF Models for Videos with Various Popularity Evolution Patterns (mRSE and 95% confidence intervals, t_r=7, t_t=30).

Videos in the Random Dataset				
Popularity Class	**Number of Videos**	**S-H Model**	**ML Model**	**MRBF Model**
Overall	16123	0.2382 ± 0.0038	0.2022 ± 0.0043	$\mathbf{0.1892 \pm 0.0032}$
Memoryless	3449	0.3351 ± 0.0099	0.2929 ± 0.0120	$\mathbf{0.2641 \pm 0.0111}$
Viral	11504	0.2086 ± 0.0040	0.1788 ± 0.0041	$\mathbf{0.1713 \pm 0.0040}$
Junk	222	0.4588 ± 0.0283	$\mathbf{0.1402 \pm 0.0274}$	$\mathbf{0.1268 \pm 0.0319}$
Quality	948	0.1921 ± 0.0124	$\mathbf{0.1707 \pm 0.0283}$	$\mathbf{0.1490 \pm 0.0199}$
Videos in the Top Dataset				
Popularity Class	**Number of Videos**	**S-H Model**	**ML Model**	**MRBF Model**
Overall	5813	0.2121 ± 0.0074	0.1837 ± 0.0081	$\mathbf{0.1723 \pm 0.0071}$
Memoryless	5130	0.2104 ± 0.0081	$\mathbf{0.1820 \pm 0.0088}$	$\mathbf{0.1740 \pm 0.0084}$
Viral	401	0.3096 ± 0.0234	0.2612 ± 0.0228	$\mathbf{0.1845 \pm 0.0242}$
Junk	78	$\mathbf{0.0560 \pm 0.0226}$	$\mathbf{0.0360 \pm 0.0236}$	0.0929 ± 0.0358
Quality	204	$\mathbf{0.1236 \pm 0.0314}$	$\mathbf{0.1315 \pm 0.0330}$	$\mathbf{0.1373 \pm 0.0479}$

Model Equations for Random Dataset:

$$\text{S-H model}: \hat{N}(v,7,30) = 1.92 \cdot N(v,7)$$

$$\text{ML model: } \hat{N}(v,7,30) = 1.22 \cdot x_1(v) + 1.24 \cdot x_2(v) + 1.36 \cdot x_3(v) + 1.52 \cdot x_4(v) + 2.23 \cdot x_5(v) + 2.33 \cdot x_6(v) + 6.15 \cdot x_7(v)$$

Model Equations for Top Dataset:

$$\text{S-H model}: \hat{N}(v,7,30) = 1.41 \cdot N(v,7)$$

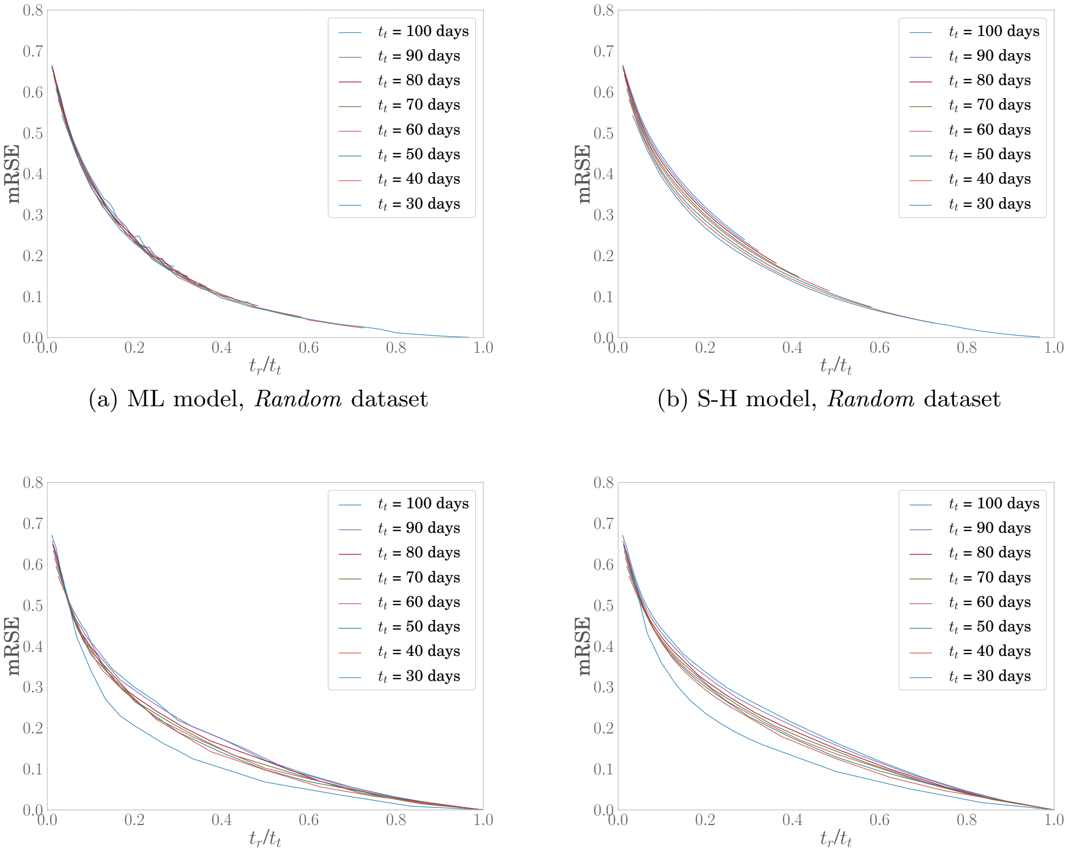
$$\text{ML model: } \hat{N}(v,7,30) = 1.19 \cdot x_1(v) + 1.02 \cdot x_2(v) + 1.16 \cdot x_3(v) + 1.36 \cdot x_4(v) + 1.35 \cdot x_5(v) + 1.47 \cdot x_6(v) + 4.82 \cdot x_7(v)$$

diction error, even if for a small fraction of the videos, may have significant impact on infrastructure management and sizing decisions as well as on the effectiveness of popularity-based information retrieval services [8], particularly given that the predictions made by the original S-H model are great overestimations, as we verified experimentally.

Junk videos are characterized by a sudden, high peak in the daily number of views followed by a steep decrease. Through some external mechanism (e.g., being featured in national news, or trending on a service like Twitter), these videos attract a burst of popularity at some point in time, but they are not *interesting* enough to continue being popular in the long run. By analyzing Junk videos in our two datasets, we found that around 90% of the videos classified as Junk in both datasets have their popularity peak at most 3 days after upload. By assigning lower weights to the first days in the range used as input (see model equations at the bottom of Table 1), the ML model produces much better predictions for such videos, compared to the S-H model. In other words, the ML model recognizes that a video that is still receiving a significant number of views after 7 days in the system is much more likely to continue receiving even more views after 30 days than a video that received a large number of views in the first two or three days with very few daily views afterwards. For *Junk* videos in the top dataset, the mRSE produced by the S-H model is already very small, leaving little room for improvements. Thus, for that dataset, both models produce results that are statistically tied.

Figures 2(a-c) allow us to determine the minimum number of daily samples (i.e., minimum t_r) so as to keep model errors below a given threshold, for the given target date t_t. For example, for t_t=30, we need to set t_r equal to at least 8 to keep the mRSE of our ML model below 0.2, for videos in the random dataset. We now extend this analysis, considering various values of t_t, and plotting, in Figure 3, mRSE results as function of the fraction of the total period up to the target date that must be monitored and sampled (i.e., $\frac{t_r}{t_t}$).

Figure 3-a) shows that our ML model is very insensitive to variations in the target date, that is, it is able to maintain the same average prediction error while still predicting longer into the future, provided that the same fraction of popularity history is sampled. The same is true, to some



(a) ML model, *Random* dataset

(b) S-H model, *Random* dataset

(c) ML model, *Top* dataset

(d) S-H model, *Top* dataset

Figure 3: Model Prediction Errors (mRSE) as Function of Fraction of History Monitored and Sampled.

extent, for the S-H model, although, having fixed the desired average prediction error, Figure 3-b) shows a slight trend towards an increase in the required fraction of history sampled as we increase the target date. This trend is more clear, for both models, in the Top dataset, as shown in Figures 3(c,d). We conjecture that the reasons for this different behavior are related to the distribution of popularity evolution patterns among the videos in both datasets, shown in Table 1 (column "Number of Videos"). Note that the vast majority of the videos in the Top dataset (88%) are in the *Memoryless* class. The popularity curves of these videos have no marked peak but rather some random fluctuations. The absence of some clear trend makes it harder to make accurate predictions. Thus, having fixed the fraction of history to be sampled, the further into the future the target date is, the harder the prediction becomes. In contrast, almost 70% of the videos in the Random dataset are classified as *Viral*, corresponding to videos with a clear trend in the popularity curve, which makes both models, and particularly our ML model, less sensitive to variations in the target date, for a fixed fraction of history sampled.

5.2 MRBF Model Results

Finally, we evaluate our extension of the ML model to include RBF features, the MRBF model. As described in Section 3.4, there are three main parameters we need to set in order to instantiate this model. The first, and simplest, is the number of RBF features to include. The two remaining parameters are σ, which controls the sphere of influence for each RBF feature, and α, which controls the tradeoff between minimizing the prediction error and minimizing the norm of the model parameter vector.

We used grid search to choose the best values for these parameters for our datasets. For each fold, the training set was split into a training subset and a cross-validation set. For each parameter, we selected a range of possible values. We then tested every combination of values for the parameters by training the model on the training subset and evaluating it on the cross-validation set. Afterwards, we selected the optimal values for the parameters as the ones that minimized the average error on the cross-validation set across all folds. Finally, we applied the model trained with these values to our test set.

For the number of RBF features, we considered using 100, 200, 500 and 1000 centers. The centers for each of these features were videos chosen uniformly at random from the training set. Suprisingly, we found very little variation in prediction error between these numbers, with all values being statistically equivalent. We report the numbers for the scenario where 100 RBF features are used, as using fewer centers leads to fewer model parameters and thus lower computational cost.

For α, we considered the values 10^0, 10^{-1}, 10^{-3}, 10^{-4}, 10^{-5}, 10^{-6}, and 10^{-9}. For σ, we first considered all non-negative integer multiples of 100 up to 1000, in an initial pass $(0, 100, 200, 300, \ldots, 1000)$. After observing the most promising value of σ among those, we performed a finer-grained search near it.

Table 1 shows the mRSE for MRBF model on both datasets, compared to the ML and the S-H models. Once again, we choose to present only the results for $t_t = 30$ days and $t_r = 7$ days, as the results for other target and reference dates are qualitatively similar. Table 2 shows the optimal parameter values for the model found through grid search for this scenario.

Table 2: Optimal values for the model parameters found by the grid search ($t_t = 30$, $t_r = 7$)

Parameter	Random Data Set	Top Data Set
# centers	100	100
σ	10.0	390.0
α	10^{-5}	10^{-9}

In both datasets, the MRBF model outperforms the ML model by a statistically significant amount (as measured by a paired t-test[10]). The reduction in mRSE compared to ML is of about 6% on both datasets. Compared to the S-H model, the reduction in mRSE reaches 21% and 19% on the Random and Top datasets, respectively.

One important question is where these gains in prediction accuracy come from. In order to shed some light on this issue, we measured how well the ML and MRBF models behave with respect to the popularity evolution patterns proposed by Crane & Sornette[5]. These results are also shown in Table 1.

In both datasets, we noticed a significant reduction in prediction error for the second largest class (Memoryless in the Random dataset and Viral in the Top dataset). We believe this behavior can be explained by the global model being biased towards the largest class, and the second largest class still being large enough that a reasonable number of videos of that class were chosen as centers of RBF features, thus allowing the MRBF model to better handle those videos. In both datasets, the reduction in prediction error for videos of the largest class was below the overall average.

For Quality videos, we noticed no reduction in prediction error in the Top dataset and a somewhat large but not statistically significant reduction in prediction error in the Random dataset. For Junk videos, we noticed an insignificant reduction in prediction error in the Random dataset and a very large *increase* in prediction error in the Top dataset. Junk and Quality videos are the two smallest classes in our datasets and, as such, we expect that very few videos of these classes were chosen as centers, thus limiting the ability of the model to deal with these videos effectively. However, these classes still have lower average relative squared errors than Memoryless and Viral videos.

5.3 Model Specialization

Last, we investigated whether building specialized models for specific categories of videos helps boosting model performance, reducing prediction errors. This was motivated by a conjecture, raised by Szabo and Huberman [15], that videos in different YouTube categories (e.g., "Comedy", "Education", "Music", "News and Politics", etc) might have different popularity profiles, and thus might benefit from specialized models. The authors raised this question but left it open.

To test this conjecture, we built specialized MRBF models for each video category. The parameters for each such specialized model were learned based on training examples from that category only. We compared the performance of the specialized models against that of a global model, built using training examples from all categories. We found, however, that the mRSE results of the specialized models were statistically tied with those produced by the global model for all categories in both datasets, with only one exception. That exception was the "Music" category in the Top dataset: videos in that category have an mRSE of 0.2756 ± 0.0326 in the global model, but only 0.2295 ± 0.0409 when predicted using the specialized model.

The reason for these results is related to the distributions of video popularity evolution patterns across different video categories. For the Random dataset, we found that the distribution of video popularity evolution patterns is about the same for all categories: the large majority of the videos are *Viral*, followed by a sizable but noticeably smaller number of *Memoryless* videos, and fewer *Quality* and *Junk* videos. Since most categories have videos with the same overall patterns, building specialized models does not lead to significant improvements. The same conclusion holds for the Top dataset: almost all videos there are *Memoryless*, regardless of the category, with smaller numbers of *Viral*, *Junk* and *Quality* videos. The Music category is an exception in that it has an even higher fraction of *Memoryless* videos than other categories: 97% of its videos, compared to 89% on the entire dataset. We believe that this difference in the distribution of popularity patterns explains why we observed a reduction in mRSE for this category.

6. CONCLUSIONS AND FUTURE WORK

The rate at which new content is uploaded to YouTube, and to other Web applications in general, has reached unprecedented marks. Whereas some pieces of content become instant hits, others experience a growing increase of attention, and the vast majority are of limited interest. In face of such a very uneven popularity distribution and heterogeneous popularity evolution patterns, system administrators may greatly benefit from accurate predictions of content future popularity to guide their infrastructure sizing and management decisions and online marketing strategies as well as improve the effectiveness of information retrieval services.

We here proposed two new models for predicting the future popularity of a given content using, as input, daily samples of the content's popularity measures up to a given reference date. We have tested our models on two datasets of YouTube videos, comparing their accuracy against that of a state-of-the-art baseline approach that predicts future popularity based only on the aggregate popularity up to the same reference date. We found that, by assigning different weights to different popularity samples within the monitoring period, our model is able to better distinguish between videos with different popularity evolution patterns, which leads to significant reductions in average prediction errors. We also found that by exploring the similarity between the video and known examples from the training set through RBF functions, we are able to reduce prediction errors even further, by adapting the prediction to better handle some specific kinds of videos.

Other avenues we intend to pursue include investigating the impact on prediction accuracy of introducing other variables to our models. For example, other sources of information about the video itself could be exploited for that purpose. Indeed, we experimented with variables such as daily samples of number of comments, number of ratings and number of users who "favorited" the video, as these pieces of information are also available in the video statistics panel. However, since these measures tend to be highly correlated with the number of views, introducing them to a linear regression model does not help. Alternatively, information related to the *user* who posted the video, such as number of subscribers, number of friends, might be more useful. In fact, Rodrigues *et al.* [13] noticed that user information is a major factor in explaining why duplicate copies of the same video end up having different popularities. Thus, we intend to investigate whether introducing user-related variables to our model improves its accuracy.

Our MRBF model was a first step at considering the popularity evolution pattern of the video at prediction time. There are other possible approaches for this task, though. A different approach is to explore the problem of directly predicting the popularity evolution pattern that a video will follow based only on observations up to the reference date. With this information, it would be possible to use different popularity prediction models for each pattern. We believe this can lead to reduced prediction errors due to exploring the differences between patterns in a more explicit form than what the MRBF model does. Thus, we believe that is the main line for future work, and the one we intend to pursue.

ACKNOWLEDGEMENTS

This research is partially funded by the Brazilian National Institute of Science and Technology for Web Research (MCT/CNPq/INCT Web Grant Number 573871/2008-6), and by the authors' individual grants from CNPq, CAPES and Fapemig.

7. REFERENCES

[1] Alexa.com. Alexa top 500 global sites. http://www.alexa.com/topsites, 2011. [Online; acessed 2-November-2011].

[2] S. Boyd and L. Vandenberghe. *Convex optimization.* Cambridge University Press, 2004.

[3] M. Cha, H. Kwak, P. Rodriguez, Y. Ahn, and S. Moon. Analyzing the video popularity characteristics of large-scale user generated content systems. *ACM Trans. on Networking*, 17(5), 2009.

[4] M. Cha, H. Kwak, P. Rodriguez, Y.-Y. Ahn, and S. Moon. I tube, you tube, everybody tubes: analyzing the world's largest user generated content video system. In *Proc. ACM Internet Measurement Conference*, 2007.

[5] R. Crane and D. Sornette. Robust dynamic classes revealed by measuring the response function of a social system. *Proc. National Academy of Sciences*, 105(41), 2008.

[6] F. Figueiredo, F. Benevenuto, and J. Almeida. The tube over time: Characterizing popularity growth of youtube videos. In *Proc. Conference of Web Search and Data Mining* , 2011.

[7] J. Friedman, T. Hastie, and R. Tibshirani. *The elements of statistical learning*, volume 1. Springer Series in Statistics, 2001.

[8] M. Gonçalves, J. Almeida, L. Santos, A. Laender, and V. Almeida. On popularity in the blogosphere. *Internet Computing*, 14(3), 2010.

[9] V. Heffernan. Uploading the avant-garde. http://www.nytimes.com/2009/09/06/magazine/06FOB-medium-t.html, 2009. [Online; acessed 2-November-2011].

[10] R. Jain. *The art of computer systems performance analysis.* John Wiley & Sons, 2008.

[11] J. Lee, S. Moon, and K. Salamatian. An approach to model and predict the popularity of online contents with explanatory factors. In *Int'l. Conf. on Web Intelligence and Intelligent Agent Technology*, 2010.

[12] K. Lerman and T. Hogg. Using a model of social dynamics to predict popularity of news. In *Proc. World Wide Web Conference.* ACM, 2010.

[13] T. Rodrigues, F. Benevenuto, V. Almeida, J. Almeida, and M. Gonçalves. Equal but different: A contextual analysis of duplicated videos on youtube. *Springer Journal of the Brazilian Computer Society*, 16(3), 2010.

[14] F. Suchanek, G. Kasneci, and G. Weikum. Yago: a core of semantic knowledge. In *Proc. World Wide Web Conference*, 2007.

[15] G. Szabo and B. Huberman. Predicting the popularity of online content. *Communic. of ACM*, 53(8), 2010.

[16] F. Wu and B. Huberman. Novelty and collective attention. *Proc. National Academy of Sciences*, 104(45), 2007.

[17] T. Wu, M. Timmers, D. De Vleeschauwer, and W. Van Leekwijck. On the use of reservoir computing in popularity prediction. In *Proc. Conference on Evolving Internet*, 2010.

[18] P. Yin, P. Luo, M. Wang, and W.-C. Lee. A straw shows which way the wind blows: Ranking potentially popular items from early votes. In *Proc. WSDM 2012*, 2012.